

# PROJECT WEREWOLF

## First-Person Shooter

[Demo](#) - [Unity project](#) - [Game files](#)

### SUMMARY

---

Project Werewolf: Day of the Rope  
A /pol/ videogame project

The latest Antifa / BLM riots are showing us how accurately The Turner Diaries predicted our current situation. Pierce's novel warned us about the jewish takeover of America, and its story of white resistance provides a detailed example for those of us living through the events he predicted to follow. In the interest of educating younger generations on the danger they face and what to do about it, our group is making a videogame based on The Turner Diaries and other white nationalist literature. With the current political climate allowing for the release of a mainstream videogame that features an interview with Yuri Bezmenov (CoD: Cold War), we feel the time is right for more explicitly pro-white interactive media.

Although we are still in the early stages of development, a playable demo of the game is available as a basic proof of concept (see the link above). We are developing the game in Unity and currently have "template" assets for every intended feature (including dialogue, missions, a save system, basic FPS mechanics, animations and models) which greatly reduces both the development time and the amount of programming required. Anyone interested in helping needs only basic computer literacy and a good sense of aesthetics, but programming knowledge is a bonus. More information about the game is provided below, and any further questions or comments may be directed at [project\\_werewolf@tutanota.com](mailto:project_werewolf@tutanota.com)

### STORY

---

#### THEME - Immersive simulation

- Simple first-person shooter loosely based on The Turner Diaries
- Linear mission design with basic NPC interaction
- Minimal dialogue with cutscenes before & after each mission

#### SETTING - America, 20 years from now

- Urban environments use architectural / city planning models
- Natural environments include terrain & assets from Unity

- Some building exteriors can be "entered" to load interiors

### PLOT - "The Day of the Rope"

- Basic dramatization of events in The Turner Diaries
- Missions recreate important scenes from the book
- Dialogue roughly follows the corresponding narrative

### NARRATIVE - Cutsscenes & dialogue

- Main characters have simplified dialogue from the book
- Random NPC dialogue focuses on jewish conspiracy & white resistance
- Starting with introductory topics (race realism, liberal ideology)
- NPCs discuss progressively more accurate information (jews, the NWO)
- Some NPCs have a dialogue option to accept bonus missions
  - Example: Supplies
  - NPC: "Help us raid the local antifa commune for supplies."
  - Player: (1) "Let's go." / (2) "I'm busy."
  - If (1), add "Raid the Commune" to mission objectives

### MISSIONS - Accelerationist

#### Phase 1 - Dialogue

- Find the other members of your cell
- Dialogue option to discuss "resistance"
- NPC responds positively or negatively
- May ignore, attack or give missions

#### Phase 2 - Stealth

- Contacts: shop owners, criminals
- Missions: sabotage, theft, kidnapping

#### Phase 3 - Combat

- Contacts: veterans, militia members
- Missions: terrorism, assassination

### NPCs - Fictional characters

- Main characters based on those in The Turner Diaries
- Random NPCs / dialogue topics:
  - Civilian - nationalism, identity politics, QAnon
  - Patriot - white genocide, cultural marxism
  - Insider - the deep state, blood libel, luciferians

## GAMEPLAY

---

### LEVEL DESIGN

- Towns & exteriors = semi-enclosed
- Mission areas = linear (start to finish)

## MISSION DESIGN

- Tree diagrams for missions & dialogue
- Branches for objectives / dialogue choices
  - Dialogue > NPC behavior
  - Objective > NPC action / scripted event

## SCRIPTED EVENTS

- Dialogue System handles cutscenes & events
- Certain game objects have scripted behavior

## ASSETS

---

|                                      |              |
|--------------------------------------|--------------|
| Environments - Sketchup              | _ integrated |
| Objects - Unity assets               |              |
| Animation - Mixamo                   | _ integrated |
| Characters - Fuse                    |              |
| Lipsync - Lipsync Pro                |              |
| Dialogue - Dialogue System for Unity | _ integrated |
| Shooting & AI - Realistic FPS Prefab |              |
| Voiceover - Natural Readers          |              |

## UNITY WORKFLOW

Install Unity 2017.4 from mega archive. Download all assets

Import Realistic FPS Prefab, Dialogue System & LipSync Pro. Add Dialogue System support & LipSync support from "RFPS 3<sup>rd</sup> party" folder in hierarchy

Add tags & layers [top right of Inspector window] according to RFPS manual. Select edit > project settings > input. Set input according to Dialogue System manual

## NPC WORKFLOW

Create model in Adobe Fuse, send to Mixamo. Enable blendshapes & download. [Optional - add animations]. Import .FBX to unity

Set shaders to diffuse detail. Set rig to humanoid. Add idle/walk/run/shoot animations. Set materials to external [legacy]

Place SoldierGoodNPC from "RFPS prefabs" folder into scene. Delete all child objects in hierarchy. Drag Mixamo object onto Soldier. Set 'object with anims' to Mixamo. Set layer to 'NPCs' & tags to 'flesh'

Add 'Soldier' controller to Mixamo. Set avatar to Mixamo object

Select GameObject > 3D > ragdoll. Drag Mixamo child objects to menu. Add LocationDamage script to each child. Set AI Component to Mixamo

## LIPSYNC WORKFLOW

Open Audacity & set recording device to Windows WASAPI. Enter desired text into <http://naturalreaders.com>, press 'record' in Audacity & play in Natural Readers. Save as .WAV & import to Unity

Select tools > Rogo > clip editor. Select settings [gear] > autosync. Set path for SOX & verify.

Add component to Mixamo object. Set blend system to blendshapes. Select Body mesh & audio source. Select 'sliders' icon > Mixamo.

Configure Dialogue System to use imported .WAV [see tutorials]

## ENVIRONMENT WORKFLOW

Create / download models in sketchup, save as 2015 version. Import .SKP to unity & enable colliders, normals, & lightmap. Set object to static (top right of Inspector).

Select window > occlusion culling & bake occlusion. Select window > navigation & bake NavMesh

## MODELS - Fuse / Sketchup / Unity

Environments - Sketchup

Characters - Fuse

Objects - Game Asset

## ANIMATION - Mixamo / Lipsync Pro

Bodies - Mixamo (upload to auto-rig)

Faces - Lipsync Pro

Objects - Unity physics

## TEXTURES - Low-detail

<https://www.textures.com/>

Edit with GIMP

## SOUNDS - Dynamic

Voices - <https://naturalreaders.com/>

Effects - <https://www.bfxr.net/>

Add Unity 3D doppler effect

## MUSIC - Martial & vaporwave

Legionarii, Waffenruhe, Triarii, L'Effet C'est Moi

Xurious, CyberNazi, Storm King, Hiraeth

## RESOURCES

---

Unity 2017.4 (works with project assets) - [Download Backup](#)

### Required Assets -

Realistic FPS Prefab 1.45 - [Download Backup](#)

LipSync Pro - [Download Manual](#)

Dialogue System for Unity - [Download Manual](#) [RFPS setup](#) [LipSync setup](#)

Adobe FUSE - <https://www.adobe.com/products/fuse.html>

Sketchup Pro 2017 - [Download](#)

## MIXAMO TUTORIAL

First, start a new project and a new scene to test your setup. I know, I know, you want to use your awesome scene that you build you drop your character in and see all the shaders in the awesome glory of your game. Sorry, spare yourself hours or days, it's just to test your character out, so new project, new scene.

1) There's a T-pose pose in Mixamo, apply that pose first. If your model isn't recognized by Mixamo's auto-rigging, hire someone on upwork first to fix the model and put into a T-Pose. Great, download that and only that pose with the skin. If you don't have a T pose, give up now, it's a lost cause. Don't think, "well, it's close, I'll fudge it cause I try other stuff on the internet and I get close and it works". You need the T-pose.

Pro Tip: If Mixamo doesn't recognize your FBX file, try an OBJ format instead.

2) Import that \*exact\* FBX model you downloaded from mixamo into your project and add it to your scene. Not any other models you have, you need that exact model. Don't try the old switcharoo here, it won't work. The model should have the animator component added to it, if it doesn't add it manually. It will likely show up without materials applied, you can add them back later, just skip it for now. It must be the model you downloaded from Mixamo with the T-Pose avatar from Mixamo. You can't mix and match, it must be these two and nothing else.

3) In your project window, click on the little arrow next to the model you added. You'll see an avatar down towards the bottom. Drag this avatar into the avatar slot on the animator component for the model in your scene. It might already be there, but if not, add it manually. It must be that exact avatar and that exact model. You can't get slick here either and try brute forcing a mix of models and avatars, it won't work.

4) Click on "Select" on the model you imported. Click on "Rig". Change the dropdown to Humanoid. Click Apply. Wait, what? It worked... I know, you're expecting the errors right? You've been here before and it didn't work with the other model you were playing around with, I know, I know. If you followed the steps above, it worked without the yellow or red errors that ruined your prior weekend.

5) In your Asset folder for the project where your model is stored, create an animation controller (right click, Create, Animation controller). Drag that controller to the open controller slot on your model.

6) Go back to Mixamo. Your model should still be loaded there, now select the animation you want, like Walk. Download that animation. Use the import function in Unity to import that into the folder where your model is stored.

7) Ah ah ah, don't add that animation yet. I know what you were thinking, it won't work. Okay, so click on the animation that you just imported into your project folder. In the inspector window, click on Rig, now change that from Generic to Humanoid and click apply. If you followed the steps correctly, no red or yellow errors here.

8) In your model, double click on the animation controller you added earlier. Drag and drop the clip you just downloaded and set to humanoid into the animator.

9) Click on your model that you added to your scene in the Hierarchy. Click Apply Root Motion, set update mode to Animate Physics.

10) To add additional animations, follow steps 6 to 8.

Optional depending on your project and animations:

11) Add Rigidbody to your model, under constraints, for Rotation, check X and Z.

12) Add a collider. Depending on your model, capsule or box collider may work best. Mesh causes a lot of additional problems, finish this step and you can tweak it later on.

13) Add a couple of cubes to your scene to test, one for the floor and one for a wall.

14) Place your model on top of the floor and in front of the wall.

15) When you play your game, your character should not fall through the floor or walk through the wall.

16) Go back and painfully add your materials to the model. You'll have to import them and then apply them. If you're lucky and placed your textures one folder above where you stored your model, they might auto-apply. If not, just put them back so your model looks complete again.

If you've done everything above, your model won't fly into space, warp backwards during animation clips, sink through the floor, or just fail entirely. You can now copy the files into your real project and tweak from there for navigation meshes, etc.

TL:DR - import .FBX > add to scene > inspector > materials > shader > legacy > diffuse

## UNITY OPTIONS

Remove gloss:

- window > lighting > env. reflections > custom > none

If stuck at "5/11 jobs":

- reduce scale to 0.1 > check static > return to 1.0 scale > generate navmesh

If lighting recalculates:

- lighting > untick auto-generate

Asset store:

- cloudstoy, LOD

## CODE

---

Load scene:

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class exit1 : MonoBehaviour{
void OnTriggerEnter(Collider collider)
{
    if (collider.gameObject.tag == "Player")
    {
        SceneManager.LoadScene ("city", LoadSceneMode.Single);
    }
}
}

```

Play audio on collision:

```

using UnityEngine;
[RequireComponent(typeof(AudioSource))]

public class voice : MonoBehaviour {
    AudioSource audioData;

    void OnTriggerEnter(Collider collider) {
        if (collider.gameObject.tag == "Player") {
            audioData = GetComponent<AudioSource>();
            audioData.Play(0);
        } else {
            audioData.Pause();
        }
    }
}

```

Click to play audio:

```

using UnityEngine;
using System.Collections;

public class ClickAudio : MonoBehaviour {

    public AudioSource Source;
    public AudioClip Clip;
    public GameObject MYOBJECT = null;
    bool displayGUI = false;

    void OnGUI() {
        if (Input.GetMouseButtonUp(0)) {
            Ray ray =
Camera.main.ViewportPointToRay(Camera.main.ScreenToViewportPoint(Input.mousePosition));
            RaycastHit outinfo;

```

```

    if (Physics.Raycast(ray, out outinfo, Mathf.Infinity)) {
        //HIT - Show GUI
        if (Physics.Raycast(ray, out outinfo, Mathf.Infinity)) {
            if (outinfo.transform.gameObject == MYOBJECT) {
                displayGUI = true;
            }
        }
    }
}
if (displayGUI == true) {
    Source.Play();
    if (GUI.Button(new Rect(10, 70, 150, 30), "Pause")) {
        Source.Pause();
        Debug.Log("Pause: " + Source.time);
    }
    if (GUI.Button(new Rect(10, 170, 150, 30), "Continue")) {
        Source.UnPause();
    }
}
}
}
}
}

```